**MINI PROJECT**

**(2020-2021)**

**IOT BASED INTELLIGENCE TRAFFIC MANAGEMENT SYSTEM**

**MID-TERM REPORT**

****

**Institute of Engineering & Technology**

**Submitted by YOGESH SARASWAT**

**(171500403)**

**SHIMANSHU SHARMA**

**(181599003)**

**VIPIN SHARMA**

**(171500383)**

***Supervised By: -***

**Mr. AMIR KHAN**

Technical Trainer

**Department of Computer Engineering & Applications**

**Content**

**DENSITY BASED TRAFFIC LIGHT CONTROLLER USING ARDUINO**

1. **About this project**
2. **Working of the project**
3. **Algorithm of a project**
4. **Component required**
5. **Layout diagram**
6. **Code explanation**
7. **Objective of the project**

**AUTOMATIC STREET LIGHT CONTROLLER USING ARDUINO**

1. **About the project**
2. **Hardware components required**
3. **Hardware connection**
4. **Layout diagram**

**About the project:**

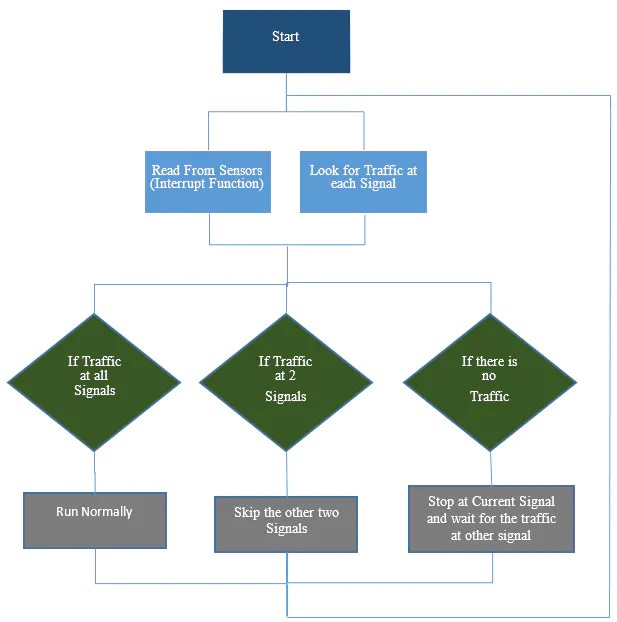
1. **Arduino is the main part of this project and it will be used to read from ultrasonic sensor HC-SR04 and calculate the distance. This distance will tell us if any vehicle is near the signal or not and according to that the traffic signals will be controlled.**
2. **The main task was to avoid use of delay because we have to continuously read from the ultrasonic sensors and also at the same time, we have to control signals which requires the use of delay function.**
3. **So we have used the timerone library which is used to repetitively measure a period of time in microseconds and at the end of each period, an interrupt function will be called. in this function, we will control the traffic signals.**

**Working of the project:**

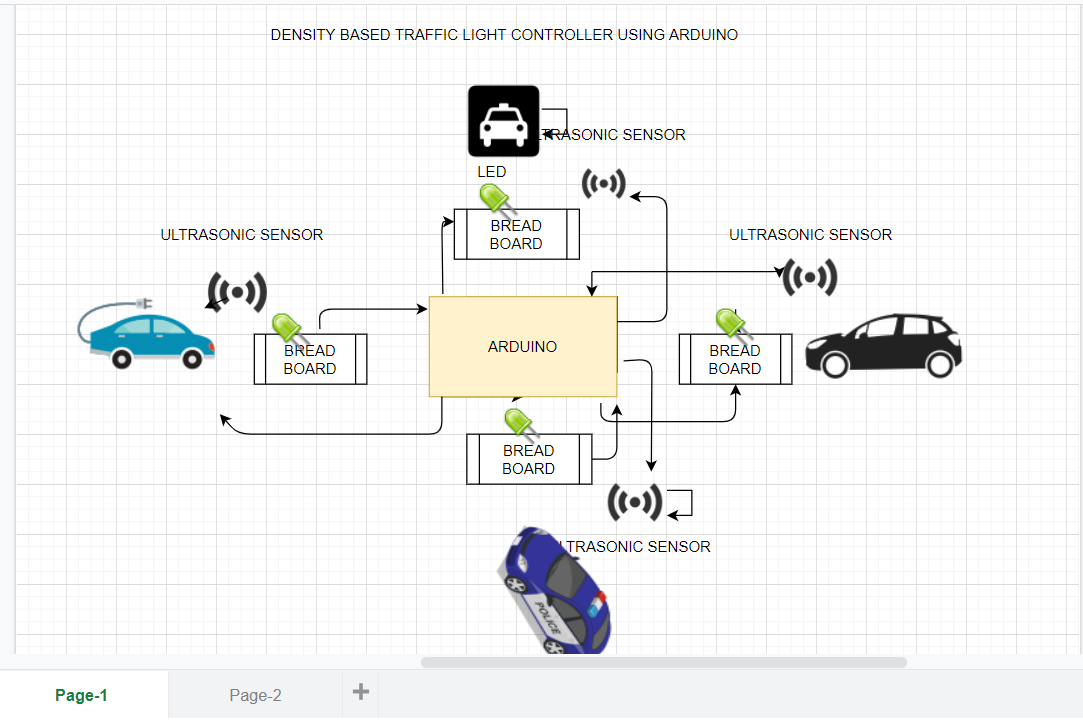
**The working of the project is divided into three steps:**

1. **If there is traffic at all the signals, then the system will work normally by controlling the signals one by one.**
2. **If there is no traffic near a signal, then the system will skip this signal and will move on to the next one. For example , if there is no vehicle at signal 2,3 and currently the system is allowing vehicles at signal 1 to pass. Then after signal 1, the system will move on to signal 4 skipping signal 2 and 3.**
3. **If there is no traffic at all the 4 signals, system will stop at the current signal and will only move on the next signal if there will be traffic at any other signal.**

**Algorithm**

****

**Layout of a project:**

****

**Explanation of a layout:**

1. **Four ultrasonic sensors are interfaced with arduino. Arduino will read from the sensors and will calculate the distance. This sensor can measure from 2 to 400 cm.**
2. **Ultrasonic sensor basically emits an ultrasonic wave from the trigger and it is received by the echo after deflecting an object. In order to generate a wave, we will have to set the trigger at high for 10 us which will send an 8 cycle sonic burst at 40 KHz which will hit the object and after hitting the object, the wave will be received by the echo. The echo will then tell us the time that the wave have travelled in us(microseconds). We will then convert this time into distance travelled by using:**

**S = v\*t**

1. **LED’s are connected to the arduino through the 220 ohm resistors. It is necessary to use the resistor with the LED. The resistor limits the current flowing through the LED. If we won’t use it then the LED will burn out soon. We can use the resistor of value from 100 ohm to 10k ohm with the LED. Larger the value of LED, lesser the current will pass.**

**Components required:**

1. **Arduino mega 2560**
2. **4 x HC-SR04 ultrasonic sensor**
3. **4 x Red LEDs**
4. **4 x Green LEDs**
5. **4 x Yellow LEDs**
6. **12 x 220 ohm resistors**
7. **Jumper cables**
8. **Breadboards**

**Code explanation:**

**First of all, we will include the timerone library. This library is used to repetitively measures a period of time in microseconds and at the end of each period, an interrupt function will be called.**

**We have used this library because we want to read from the sensors and control LED’s at the same time. We will have to use the delay in between the traffic signal so we can’t read from the sensors continuously. Therefore we have used this library which will allow us to call a function in which we will read from the sensors continuously and in the loop function, we will control the traffic signals.**

**#include<TimerOne.h>**

**In the setup function, we have used the Timer1.initialize(microseconds) function. This must be called before we use any of the other methods of timerone library. “Microsecond” is actually the period of time the timer takes. It is optionally to specify the timer’s period here. The default period is 1 second. Keep in mind that it breaks analogWrite() on digital pins 9 and 10.**

**Timer1.initialize(1000);**

**Timer1.attachInterrupt(softInterr) calls a function each time the timer period finishes. We have set the timer period at 100000 so our function will called after 100 millseconds.**

**Timer1.attachInterrupt(softInterr);**

**In the loop function it is looking if there is any vehicles under the 5cm distance or not. If there will be vehicle, then the function to that signal will be called.**

**Void loop ()**

**{**

**// if there are vehicles at signal 1**

**if (S1<t)**

**{**

**signal1Function ();**

**}**

**// if there are vehicles at signal 2**

**if(S2<t)**

**{**

**signal2Function ();**

**}**

**// if there are vehicles at signal 3**

**if(S3<t)**

**{**

**signal3Function ();**

**}**

**// if there are vehicles at signal 4**

**if(S4<t)**

**{**

**signal4Function ();**

**}**

**}**

**‘Softinterr()’ is the interrupt function and it will called after every 100 millisecond. In this function, we have read from the ultrasonic sensor and have calculated the distance.**

**void softInterr ()**

**{**

**// reading from first ultrasonic sensor**

**digitalWrite(trigger1, LOW);**

**delayMicrosecond(2);**

**digitalWrite(triggerpin1, HIGH);**

**delayMicrosecond(10);**

**digitalWrite(triggerpin1, LOW);**

**time = pulseIn(echopin1, HIGH);**

**S1 = time \* 0.034/2;**

**Main objective:**

**If there will be no traffic on the other signal, one shouldn’t wait for that signal. System will skip that signal and will move on next one. In this way we can save our time while travelling.**

**AUTOMATIC STREET LIGHT CONTROLLER**

**About the project:**

**Automatic street light control is used to control the street lights (turn on and off based on the light). Here we make use of LDR (light dependent resistor) and LED(light emitting diode) and arduino. When cars comes then automatically the LED will blink and after that it will stop blinking.**

**Hardware components required:**

1. **LDR**
2. **LED**
3. **4.7K register**
4. **Bread board**
5. **Connecting wires**
6. **Arduino**

**LDR is used to detect the light, arduino is used to on/off the light. It is also known as a photoresistor. These are made up of semiconductor material that have high resistance. There are many different symbols used to indicate a photoresistor or LDR, one of the most commonly symbol.**

**Layout diagram:**

****

**Hardware connections:**

1. **Arduino 3rd pin connected to LED +ve**
2. **Arduino GND connected to LED –ve through 4.7k**
3. **Arduino +5v is connected to LDR one end**
4. **Arduino A0 pin is connected to LDR other end**
5. **Arduino GND is connected to LDR other end with 4.7k.**
6. **Arduino**

**CODING WORK IS GOING ON FOR THIS STREET LIGHT CONTROLLER**